

化工应用数学 第五章 常微分方程求解 讲义

常微分方程的基本概念:

在化工研究中,很多时候我们所研究的变量之间的函数关系并不能直接写出来,而是要根据物理或者实际情况,先得到这些变量之间的导数的关系,然后再根据这些关系去把未知的函数关系求解出来。

下面介绍一些微分方程中的基本概念:

这种含有未知函数的导数的方程就称为微分方程。

如果未知函数是一元函数,那么这种含有未知函数导数的微分方程称为常微分方程;而若未知函数是多元函数,那么这种含有未知函数的偏导数的微分方程称为偏微分方程。

这里,方程中未知函数的导数的最高阶数称为微分方程的阶

例如 $\frac{dy}{dx} = 4x^2$, $\frac{d^2\theta}{dt^2} + \frac{\mu}{m} \frac{d\theta}{dt} + \frac{g}{l} \theta = 0$

一阶 二阶

一般的,二阶及二阶以上的微分方程我们称为高阶微分方程。

同时,最高阶导数的指数我们称为微分方程的次数,上面两个都是一次微分方程。

能够满足微分方程的某个函数 $y=f(x)$ 叫做微分方程的一个解。

一般的, n 阶微分方程的解并不是唯一的,我们把包含 n 个独立的任意常数,并且常数个数与微分方程的阶数相同的解称为微分方程的通解,写作:

$$y=f(x,C_1,C_2,\dots,C_n)$$

通解中各任意常数取特定值所得到的解称为特解。

常微分方程的解析求解-分离变量法:

若常微分方程 $dy/dx = \phi(x,y)$ 可以将 $\phi(x,y)$ 分解为 x 、 y 分别的两部分即:

$$\frac{dy}{dx} = f(x)g(y)$$

对于这样的常微分方程,就可以使用分离变量法进行求解,即把上式数学变换为:

$$\frac{dy}{g(y)} = f(x)dx$$

然后两边分别进行不定积分即可进行求解:

$$\int \frac{dy}{g(y)} = \int f(x)dx + C$$

例: 求微分方程 $\frac{dy}{dx} = 2xy$ 的通解。 解: $y = Ce^{x^2}$

而有些常微分方程虽然并不能像上面那样直接分解为 x 、 y 两部分,但是可

以通过一定的数学变换转换为可分解的形式，这时我们也可以先数学变换、再进行分离变量法进行求解。这种形式的常微分方程的一个代表就是齐次微分方程，即方程形式为：

$$\frac{dy}{dx} = f\left(\frac{y}{x}\right)$$

此时，可以先进行变量代换： $u=y/x$ ，即 $y=ux$

$$\frac{dy}{dx} = u + x \frac{du}{dx}$$

结合原微分方程，有：

$$u + x \frac{du}{dx} = f(u)$$

上式就可以变化为可分离变量的微分方程：

$$\frac{du}{dx} = \frac{f(u) - u}{x}$$

之后再根据 $f(u)-u$ 是否等于 0，分别进行讨论计算

若不等于 0，通解为： $x = Ce^{\varphi(\frac{y}{x})}$ 其中 $\varphi(u) = \int \frac{du}{f(u)-u}$

若等于 0，说明 $f(u)=u$ ，通解为： $\ln|x| = \ln|y| + c$

例：求解微分方程： $x \frac{dy}{dx} = \frac{3y^3+6x^2y}{3x^2+2y^2}$ **解：** $x^3 = cy\sqrt{y^2+3x^2}$

常微分方程的解析求解-一阶线性方程：

另一种都具有解析解的常微分方程是一阶线性微分方程，其具有下述形式：

$$\frac{dy}{dx} + P(x)y = Q(x)$$

当 $Q(x)$ 等于 0 时，上述方程称为齐次的

当其不等于 0 时，上述方程称为非齐次的

对于齐次情况，可以使用分离变量法：

$$\frac{dy}{y} = -P(x)dx, \int \frac{dy}{y} = -\int P(x)dx$$

$$\ln y = -\int P(x)dx + \ln C,$$

$$\text{齐次方程的通解为 } y = Ce^{-\int P(x)dx}$$

对于非齐次情况，首先计算对应的齐次情况，然后使用常数变易法求解，下面以例子来具体讲解：

线性非齐次方程 $\frac{dy}{dx} + P(x)y = Q(x)$.

讨论 $\therefore \frac{dy}{y} = \left[\frac{Q(x)}{y} - P(x) \right] dx,$

两边积分 $\ln|y| = \int \frac{Q(x)}{y} dx - \int P(x) dx,$

设 $\int \frac{Q(x)}{y} dx$ 为 $v(x)$, $\therefore \ln|y| = v(x) - \int P(x) dx$

即 $y = e^{v(x)} e^{-\int P(x) dx}$. 非齐次方程通解形式

与齐次方程通解相比 $C \Rightarrow u(x)$

■ 例: 求方程 $\frac{dy}{dx} - \frac{2y}{x+1} = (x+1)^{\frac{5}{2}}$ 的通解.

解 这是非齐次线性方程 先求对应的齐次方程的通解.

$$\frac{dy}{dx} - \frac{2}{x+1}y = 0, \quad \frac{dy}{y} = \frac{2dx}{x+1}, \quad y = C(x+1)^2.$$

用常数变易法, 把 C 换成 u , 即令 $y = u(x+1)^2$,

$$\frac{dy}{dx} = u'(x+1)^2 + 2u(x+1),$$

代入非齐次方程, 得 $u' = (x+1)^{\frac{1}{2}}.$

两端积分, 得 $u = \frac{2}{3}(x+1)^{\frac{3}{2}} + C.$

故所求通解为 $y = (x+1)^2 \left[\frac{2}{3}(x+1)^{\frac{3}{2}} + C \right].$



15

常微分方程的几何解法:

常微分方程的解析方法与几何方法

解析方法	几何方法
$y' = f(x, y)$	方向场/斜率场/线素场
$y_1(x)$	积分曲线

方向场/斜率场/线素场: 平面上, 通过每点做短线素, 线素的斜率等于每一

点的 $f(x,y)$ 值

积分曲线：平面上每一点都与线素场中的线素相切的一组曲线，其即为微分方程解的图像

定理： $y_1(x)$ 是方程 $y'=f(x,y)$ 的解的充分必要条件是 $y_1(x)$ 的图像是这一微分方程的积分曲线。

根据上述定理，画出相应的积分曲线实际上也是求解常微分方程的一种方法。

如何根据微分方程画积分曲线：

a. 计算机方法：

- 1) 平面上选择不同的点 (x,y)
- 2) 对每一点计算 $f(x,y)$
- 3) 在 (x,y) 处画斜率为 $f(x,y)$ 的线素

b. 人工方法：

- 1) 选择不同的斜率 C
- 2) 画出函数 $f(x,y)=C$ 的图像
- 3) 在上述曲线上取不同的点画出斜率为 C 的线素

例：通过几何法求解微分方程： $\frac{dy}{dx} = -\frac{x}{y}$ 、 $\frac{dy}{dx} = 1 + x - y$

对比两个函数的解析解： $x^2+y^2=C$ 和 $y=(xe^x+C)e^{-x}=x+Ce^{-x}$

常微分方程数值求解-尤拉法/欧拉法：

对于比较复杂的问题，若前面介绍的解析方法无法求解、几何方法求解也不方便时，对于给定初值的问题，我们可以使用欧拉法进行数值求解。

针对一阶初值问题：

$$\begin{cases} y' = f(x, y) & (a < x < b) \\ y(a) = y_0 \end{cases}$$

将 $[a,b]$ 进行 n 等分，所得节点记为 x_i ，对应的数值计算值为 u_i 、对应的函数值为 y_i ，记每等分为步长 $h=(b-a)/n$

将 $y(x)$ 在 x_i 附近进行泰勒展开，可得：

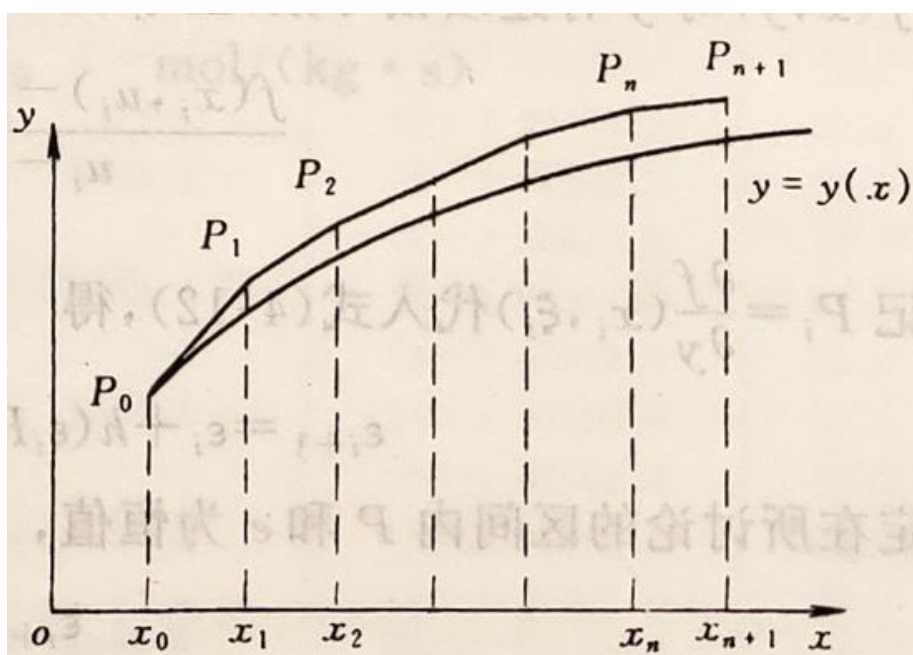
$$y(x_{i+1}) = y(x_i) + (x_{i+1} - x_i)y'(x_i) + \frac{(x_{i+1} - x_i)^2}{2!}y''(\xi_i)$$

$x_i \leq \xi_i \leq x_{i+1}$

舍去高阶小量之后，上式可写为：

$$\begin{cases} u_{i+1} = u_i + hf(x_i, u_i) & i=0, 1, 2, \dots, n-1 \\ u_0 = y_0 \end{cases}$$

这一公式就是显式欧拉公式。



上图为显式欧拉法求解的示意图，实际上显式欧拉法就是利用 x_i 处的斜率代替 (x_i, x_{i+1}) 整体上的斜率

同样的，我们也可以使用 x_{i+1} 处的斜率代替 (x_i, x_{i+1}) 整体上的斜率，这时数学上相当于将 $y(x)$ 在 x_{i+1} 附近进行泰勒展开：

$$y(x_i) = y(x_{i+1}) - h y'(x_{i+1}) + \frac{h^2}{2!} y''(\xi_i)$$

同样的，经数学处理可得：

$$\begin{aligned} u_{i+1} &= u_i + h f(x_{i+1}, u_{i+1}) \\ u_0 &= y_0 \end{aligned}$$

上式即为隐式欧拉法。

因为上式中，方程左右均含有 u_{i+1} ，所以求解过程需要进行迭代求解，此时迭代公式我们直接使用上式本身，即：

$$u_{i+1}^{(k+1)} = u_i + h f(x_{i+1}, u_{i+1}^{(k)})$$

数学上，我们可以证明隐式欧拉法是绝对稳定的，但是精度上其与显示欧拉法都只有一阶精度。为了提高精度，我们结合显式与隐式欧拉法，可以得到：

$$u_{i+1} = u_i + \frac{h}{2} [f(x_i, u_i) + f(x_{i+1}, u_{i+1})]$$

可以发现，这里我们相当于是使用 x_i 与 x_{i+1} 处的斜率的平均值代替 (x_i, x_{i+1}) 整体上的斜率，所以上式也称为梯形法则，其具有二阶精度。

龙格-库塔法：

从前面的内容，我们可以发现，通过引入更多的点来代表区域中的斜率能够

有效的提高计算的精度，基于这一思想，提出了龙格-库塔方法：

$$u_{i+1} = u_i + \sum_{j=1}^v \omega_j K_j$$

其中：

$$K_j = hf\left(x_i + c_j h, u_i + \sum_{l=1}^{j-1} a_{jl} K_l\right) \\ c_1 = 0 \\ x_i = x_0 + ih$$

这里使用 v 个节点去表征区域上的斜率，每个节点赋予了不同的权重 ω_j

可以发现，如果我们仅使用一个节点、同时赋予权重，那么龙格-库塔法就退化为欧拉法；同时也可以证明，之前所讲的梯形法则也是龙格-库塔法使用两个节点的一个特例。

使用 v 个节点的龙格-库塔公式一般称为 v 阶龙格-库塔公式，公式中各参数参数的确定是使用待定系数法，将其与函数的泰勒展开式对比得到的，下面是一般使用的精度比较好的四阶龙格-库塔法公式：

$$\begin{cases} y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = hf(x_i, y_i) \\ K_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}K_1\right) \\ K_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}K_2\right) \\ K_4 = hf(x_i + h, y_i + K_3) \end{cases}$$

外推法：

很容易能想到，缩小步长 h 也是提高精度的一个方式，但 h 的减小会显著的提高总体的计算量；同时，由于截断误差的存在（之前公式推导过程中忽略的高阶小项），过小的步长也可能会导致截断误差的累计，从而精度反而减小。

针对这一问题提出的方法即为外推法，即将多个步长的计算结果结合起来，以提高计算的精度。

比如，若使用欧拉法在步长 h 与 $h/2$ 两种步长上进行计算，由于普通欧拉法是一阶精度，误差和步长成正比，有：

$$\begin{aligned}u_i &\approx y(x_i) + \phi h \\w_i &\approx y(x_i) + \phi \frac{h}{2}\end{aligned}$$

这里我们认为 ϕ 是常数，那么通过上述两式消去 ϕ 即可得到：

$$y(x_i) \approx 2w_i - u_i$$

通过这样的计算，我们可以得到高于前两者的精度。

同样的，对于梯形法则我们有：

$$\begin{aligned}u_i &\approx y(x_i) + \phi h^2 \\w_i &\approx y(x_i) + \phi \left(\frac{h}{2}\right)^2\end{aligned}$$

从而：

$$y(x_i) \approx \frac{4w_i - u_i}{3}$$

通过这样的方式，我们就可以在不计算更小步长的情况下，提高计算的精度。
(更小的步长会带来计算量的增加以及可能的计算不稳定)

边值问题：

前面数值计算是用来解决初值问题的，这也是我们在实际常微分方程求解过程中较多遇到的问题；而实际过程中，我们还会遇到另一类的常微分方程：没有提供初值，而是提供在边界处的函数值、导数值或者两者的结合，即边值问题。

针对这类问题，常见的解决方案有两种：

1. 将边值问题转化为等价的初值问题
2. 将问题进行有限差分，仅为转化为三角矩阵的形式，从而使用追赶法进行求解。

但针对这些问题，因为这种情况一般都是对应于稳态的扩散方程等问题，所以更加推荐的方式是使用一些商业或者开源的 CFD 软件进行求解，这部分将在偏微分方程求解部分一同讲解，这里不再具体介绍。

常微分方程 Python 求解：

Python 中求解微分方程的思路有很多，这里我们使用 `scipy` 中提供的用于解常微分方程的函数 `odeint()`，其基本语法为：

```
scipy.integrate.odeint(func, y0, t, args=(), Dfun=None, col_deriv=0, full_output=0, ml=None, mu=None, rtol=None, atol=None, tcrit=None, h0=0.0, hmax=0.0, hmin=0.0, ixpr=0, mxstep=0, mxhnil=0, mxordn=12, mxords=5, printmessg=0, tfirst=False)
```

实际使用中，还是主要使用前三个参数，即微分方程的描写函数、初值和需要求解函数值对应的的时间点。这个函数使用过程中要求微分方程必须化为标准

形式，并且可以处理常微分方程组的问题。（高阶微分方程可以转换为微分方程组来解决）

例：求解 $dy/dx=x$

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def diff(y, x):# dy/dx = x
    return x

x = np.linspace(0, 10, 100) # 给出x范围
y = odeint(diff, 0, x) # 设初值为0
plt.plot(x, y[:, 0])
plt.grid()
plt.show()
```

例：求解单摆运动： $\frac{d^2\theta}{dt^2} = -\frac{g}{l}\theta$

引入角速度 ω ，将二阶常微分方程转化为一阶常微分方程组：

$$\frac{d\theta}{dt} = \omega, \frac{d\omega}{dt} = -\frac{g}{l}\theta$$

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

g = 9.8
l = 1
def diff2(d_list, t):
    omega, theta = d_list
    return np.array([-g/l*theta, omega])
t = np.linspace(0, 20, 2000)
result = odeint(diff2, [0, 35/180*np.pi], t)
# 结果是一个两列的矩阵，odeint中第二个是初始单摆角度35度
plt.plot(t, result[:, 0]) # 输出omega随时变化曲线
plt.plot(t, result[:, 1]) # 输出theta随时变化曲线，即方程解
plt.show()
```